Metoda shooting

Preliminarii

În rezolvarea unei ecuații diferențiale ordinare de ordinul n avem nevoie de n condiții inițiale specificate într-un punct x_0. În cazul în care ecuația sau sistemul care trebuie rezolvat nu are toate condițiile inițiale precizate iar o parte dintre aceste condiții din condiții sunt date pe frontieră, atunci această problemă se poate rezolva folosind *metoda shooting*. Metoda poartă această denumire, în traducere - problema tirului, deoarece este similară cu problema unui ofițer de artilerie care după ce a încadrat ținta cu două lovituri (una lungă şi una scurtă) trebuie să dirijeze tirul în aşa fel încât în cele din urmă proiectilul să atingă ținta. Bineînțeles că o înălțime intermediară primelor două va avea drept efect o lovitură mai apropiată de obiectiv. (vezi Gerald şi Wheatley, 1999).

Există în general două moduri de aplicare a metodei shooting. Prin analogie cu problema aflării rădăcinii unei ecuații algebrice primul mod este similar cu metoda înjumătățirii intervalului, iar cel de-al doilea mod cu metoda lui Newton sau metoda tangentei. În funcție de limbajul de programare în care se va implementa algoritmul se poate alege una dintre cele două variante.

Metoda shooting pentru ecuații diferențiale ordinare cu o singură condiție inițială necunoscută (*înjumătățirea intervalului*)

Presupunem că avem o ecuație diferențială ordinară care are precizate condițiile inițiale într-un punct x_0 şi condiția pe frontieră în punctul x_L.

$$f'(x,y) = 0$$

$$f(x_0) = f_0;\ \ f(x_L) = f_L$$

Vom rezolva ecuația considerând-o ca o problemă inițială în care valoarea inițială este aleasă arbitrar în punctul x_0. Corectitudinea acestei alegeri este verificată rezolvând numeric ecuația, rezolvare în care putem folosi orice metodă pentru probleme cu condiții inițiale, şi văzând cât de aproape este soluția de valoarea ei reală din x_L. De obicei, pentru o alegere arbitrară, condiția pe frontieră în x_L nu este niciodată satisfăcută. De acest motiv se folosesc diferite procedee care să ne conducă la găsirea valorii inițiale corecte.

Un astfel de procedeu, asemănător cu algoritmul de înjumătățire al intervalului pentru găsirea rădăcinii unei ecuații algebrice, este prezentat în continuare (vezi Chakraborty, 1998). Presupunem că avem o ecuație diferențială ordinară de ordinul trei cu următoarele condiții la limită:

$$f'''(f, f', f'', x) = 0$$

$$f(x_0) = \alpha_0,\ f'(x_0) = \beta_0,\ f'(x_L) = \gamma_L$$

Dorim să aflăm valoarea f “$(x_0) = \gamma_0$ care să satisfacă condiția $f(x_L) = \gamma_L$. În acest sens vom alege două valori f “$(x_0) = S_1$ şi f “$(x_0) = S_2$ astfel încât prin rezolvarea ecuației cu aceste condiții inițiale obținem două valori f ‘$(x_L) = r_1$ şi f ‘$(x_L) = r_2$ cu proprietatea că $r_1 < \alpha < r_2$. Vom rezolva apoi ecuația pentru f “$(x_0) = S$, unde

$$S = \frac{S_1 + S_2}{2}$$

şi pentru care obținem f ‘$(x_L) = r$. Dacă $r < \gamma_L$ atunci S_1 va primi valoarea S, iar dacă $r > \gamma_L$ atunci S_2 va primi valoarea S (vezi Figura 1). Prin repetarea acestui procedeu intervalul din care alegem valoarea inițială se înjumătățeşte până când f ‘(x_L) este suficient de apropiat de γ_L pentru S din acest interval.

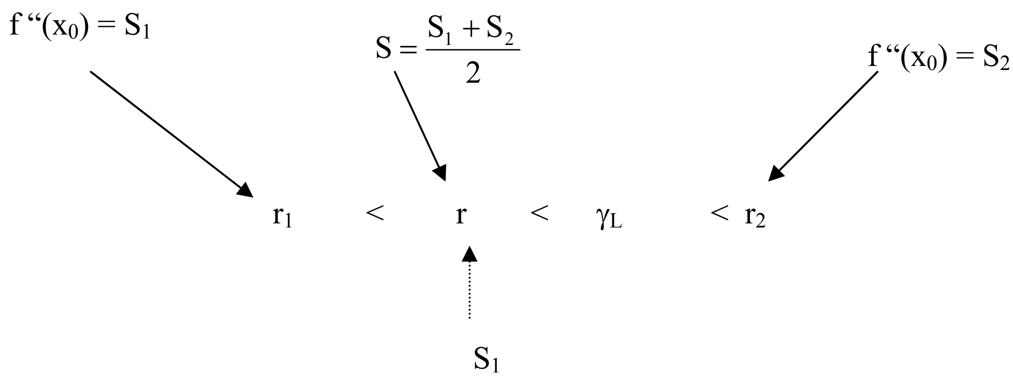

Figura 1

Modul de alegere a noilor valori S_1 şi S_2

Exemple
Rezolvarea ecuației lui Blasius folosind metoda shooting (înjumătățirea intervalului)

Considerăm ecuația lui Blasius (vezi Oroveanu, 1967), ecuație diferențială care modelează scurgerea cu strat limită pe o placă plană:

$$f''' + 0.5\, f\, f'' = 0$$
$$f(0) = 0,\ f'(0) = 0,\ f'(\infty) = 1 \qquad (1)$$

unde $f = f(\eta)$.

Pentru rezolvarea ecuației (1) considerăm două valori particulare pentru condiția inițială lipsă, $f''(0)$, şi anume $S_1 = 0$ pentru care prin rezolvarea problemei inițiale prin metoda Runge – Kutta (vezi Ixaru, 1980) obținem $f'(\infty) = 0$ şi $S_2 = 1$ pentru care vom obține $f'(\infty) = 2.085456$. Aşadar, am încadrat condiția inițială lipsă între două valori şi putem aplica în continuare algoritmul metodei shooting.
Dăm în continuare programul scris în Matlab care rezolvă această problemă. Funcția Blasius este necesară pentru aplicarea metodei Runge – Kutta (funcția ode45) care este implementată în Matlab. Variabilele care apar în program au următoarele semnificații:

a, b – capetele intervalului;
S_1, S_2 – aproximațiile pentru $f''(0)$;
yp – valoare lui f' în ∞;
k – numărul de iterații;
A – matricea rezultatelor, $A = [\eta, f'', f', f]$;

```
function yp=Blasius(x,y)
yp=zeros(3,1);
yp(1)=-0.5*y(3)*y(1);
yp(2)=y(1);
yp(3)=y(2);
```

```
format long;
a=0;
b=10;
S1=0;
S2=1;
yp=2;
k=0;

while abs(yp-1)>1e-10
   S=(S1+S2)/2;
   y0=[S,0,0];
   [x,y]=ode45('Blasius',[a:0.1:b,y0);
   [m,n]=size(y);
   if y(m,2)<1
   S1=S;
   else
   S2=S;
   end;
   yp=y(m,2);
   k=k+1;
   fprintf('nr_iter=%d ys=%f yp=%f\n', k,S,yp)
   pause;
end;

A=[x,y];
disp('  eta    fsec    fprim   f  ')
disp(A);
plot(x,y(:,2));
```

În urma rulării programului, conform Tabelului 1, se observă apropierea de soluția reală pe măsură ce numărul de iterații crește și că valoarea finală pentru f "(0) este în bună concordanță cu rezultatele obținute de alți autori, f "(0) = 0,332 (vezi Pop şi Postelnicu, 1999). Modul de lucru al algoritmului se poate urmări în Figura 7, în care curbele sunt etichetate cu iterația la care a fost obținută. Se vede că soluțiile obținute la anumite iterații prind într-un "cleşte" soluția finală, cleşte care se strânge odată cu creşterea numărului de iterații.

Numărul de iterații	f '(∞)	f "(0)
1	1.3137341	0.500000
2	0.827249	0.250000
3	1.084427	0.375000
4	0.960232	0.312500
5	1.023282	0.343750
6	0.992011	0.328125
7	1.007707	0.335938
8	0.999875	0.33203

Tabel 1
Valorile pentru f '(∞) şi f "(0) la diferite iterații

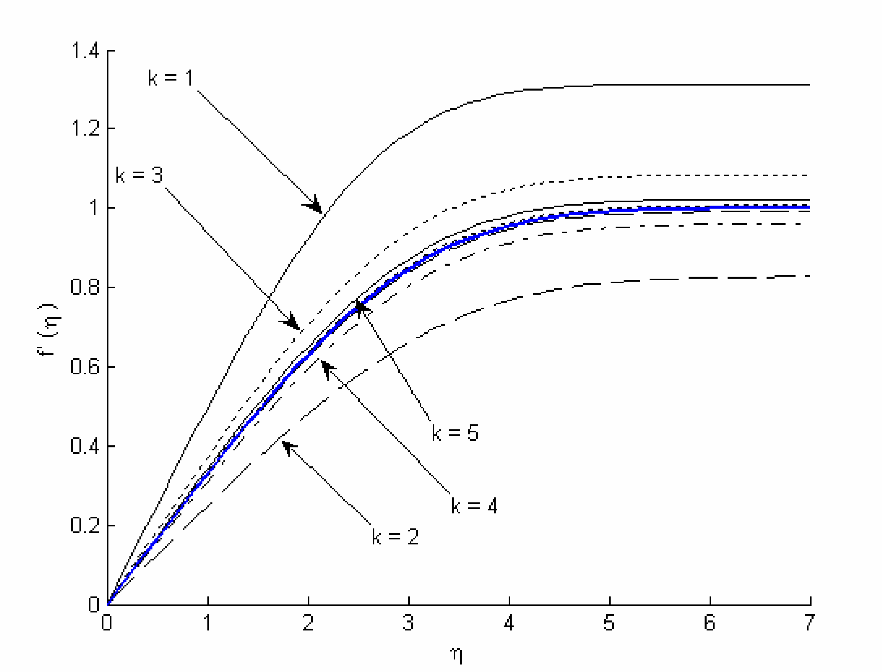

Figura 2
Aproximații succesive ale soluției.

η	$f''(\eta)$	$f'(\eta)$	$f(\eta)$	η	$f''(\eta)$	$f'(\eta)$	$f(\eta)$
0.0000	0.3320	0.0000	0.0000	4.0000	0.0642	0.9555	2.3056
0.5000	0.3309	0.1659	0.0415	4.5000	0.0340	0.9795	2.7899
1.0000	0.3230	0.3298	0.1656	5.0000	0.0159	0.9915	3.2831
1.5000	0.3026	0.4868	0.3701	5.5000	0.0066	0.9968	3.7803
2.0000	0.2667	0.6297	0.6500	6.0000	0.0024	0.9989	4.2794
2.5000	0.2174	0.7512	0.9962	6.5000	0.0008	0.9997	4.7790
3.0000	0.1614	0.8460	1.3967	7.0000	0.0002	0.9999	5.2789
3.5000	0.1078	0.9130	1.8376				

Tabel 2. Valorile pentru f, f ' şi f ''